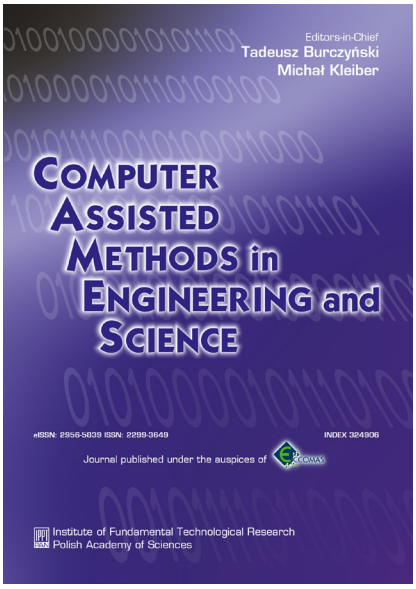
JOURNAL PRE-PROOF



Title: Metamodel Selection and Hyperparameter Tuning for Hybrid Optimisation with a Quantum-Inspired Evolutionary Algorithm in Multiple Drop Test Simulation

Author(s): Adam RURAŃSKI, Wacław KUŚ

DOI: https://doi.org/10.24423/cames.2026.1971

Journal: *Computer Assisted Methods in Engineering and Science*

ISSN: e-ISSN: 2956-5839, ISSN 2299-3649

Publication status: In press

Received: 2025-11-18

Revised: 2026-04-24

Accepted: 2026-05-14

Published pre-proof: 2026-05-19

Please cite this article as:

Rurański A., Kuś W., Metamodel Selection and Hyperparameter Tuning for Hybrid Optimisation with a Quantum-Inspired Evolutionary Algorithm in Multiple Drop Test Simulation, *Computer Assisted Methods in Engineering and Science*, 2026, https://doi.org/10.24423/cames.2026.1971

Metamodel Selection and Hyperparameter Tuning for Hybrid Optimisation with a Quantum-Inspired Evolutionary Algorithm in Multiple Drop Test Simulation

Adam RURAŃSKI[1)] , Wacław KUŚ [1)*]

[1)] Silesian University of Technology, Faculty of Mechanical Engineering, Department of Computational Mechanics and Engineering, Gliwice, Poland

*corresponding author e-mail: waclaw.kus@polsl.pl

This study introduces a hybrid optimization framework for the multi-drop testing of a lithium-ion battery enclosure. The framework integrates a Quantum-Inspired Evolutionary Algorithm (QEA) with surrogate modeling techniques. The contribution of the present study lies in metamodel selection, hyperparameter tuning, and the evaluation of two hybrid integration strategies built on a previously published QEA framework. Three types of metamodels were applied—Artificial Neural Networks (ANN), Kriging, and Polynomial Regression (PNR)—using datasets generated via Latin Hypercube Sampling and from prior QEA iterations. The hyperparameters tuning methods are the main part of the paper. Two fitness functions were analyzed, including a logarithmically scaled variant designed to compress the output range for damaged cases and enhance classification accuracy near the damage/no-damage boundary. A dual-model strategy was employed for ANN with model switching determined by a plastic strain threshold. Across the tested datasets, ANN frequently identified low-objective individuals, while PNR occasionally exhibited instability.Two hybrid schemes were implemented: HYBRID1 yielded the lowest objective values in the reported runs, whereas HYBRID2 reduced runtime and showed a steadier optimisation course, at a slight cost to the best obtained objective value. Within the analyzed QEA-based workflow, the combination of QEA, ANN, and the FF2 objective function reduced FEM computational time by an order of magnitude while maintaining decision-making effectiveness. These results support the proposed hybridization as a practical improvement of the existing QEA workflow for the studied multi-drop test problem.

Keywords: surrogate modeling; metamodel selection; hyperparameter tuning; artificial neural network; Kriging; polynomial regression; quantum-inspired evolutionary algorithm; multiple drop test; battery housing.

1. Introduction

Nowadays, highly computational tasks occur in ordinary engineering routines in industry. The standard approach to perform analysis using only Finite Element Method (FEM) and optimisation techniques, particularly in the optimisation process, is no longer sufficient. It is becoming increasingly popular to combine other techniques to improve efficiency and reduce computational costs [1–6]. Optimisation-based methods have already been successfully applied to various engineering problems, including structural, which further motivates the development of more efficient optimisation workflows [7–9]. In recent years, artificial intelligence (AI) tools have started to be implemented even in commercial industrial software such as the 3D Experience Platform [10], Ansys [11], and Altair [12]. However, these tools cannot address all existing industrial needs. Certain cases and tasks still require not general but specific, individual approach mostly based on existing tools but with addition of new scripts, interfaces or approaches to problem solution [13]. In addition to parametric optimisation tools that are already implemented, there remains the potential to reduce computational costs (defined as wall time of computations) by predicting objective function values based on a previously computed cases stored in databases or/and supported by metamodels. Such surrogate-based strategies have been shown to effectively accelerate repetitive simulations by replacing full high-fidelity models with appropriately trained reduced-order or metamodel representations [14, 15].

An example of such a problem is the optimisation of a battery housing - part exposed to multiple drop tests according to standard code[16]. Many devices used in industry but also home appliances, are at risk of dropping, prompting industry to conduct tests to ensure the intended and reliable safety performance. Devices subject to multiple drops include battery-powered tools, such as electronic equipment or garden tools. It is also important to note that batteries present an elevated level of fire and/or explosion risks. FEM tools currently offer full capabilities to perform drop test simulations and can be extended due to open APIs and scripting support. The automation of multiple drop test loops, and customized optimization tools [16] allow the solution of the optimisation process. The computational costs is high and it is important to reduce it to be possible to apply these methods in daily engineering tasks in industry.

The authors propose method based on currently published works and improved approach allowing significantly reduce computations time is proposed. To shorten computational times, the authors propose supporting the optimisation with metamodeling techniques, including Artificial Neural Networks (ANNs), the Kriging method, and Polynomial Regression (PNR). The All of these metamodeling techniques were compared and evaluated for prediction efficiency. Ultimately, the paper evaluates surrogate-assisted extensions of the previously established QEA workflow and compares two QEA-based hybrid approaches.

To provide a clear structure of this work, Section 2 describes in detail the problem formulation, objective functions, database preparation using Latin Hypercube Sampling (LHS)

and QEA approaches, prediction strategies, hyperparameter tuning procedures, and the implementation of the two proposed hybrid strategies (HYBRID1 and HYBRID2). Section 3 presents a numerical example of a battery housing drop-test model, along with database structures of various sizes and the process of tuning ANN, Kriging, and Polynomial Regression metamodels. The optimisation processes based on predictions from different metamodels are compared, and the most suitable metamodel is selected for the final implementation in the HYBRID optimisation approach. The results of the optimisation with the ANN metamodel are presented and summarized in Section 4. It should be noted that the primary goal of this study is to investigate the behaviour and efficiency of hybrid optimisation strategies combined with different metamodeling techniques within a computationally demanding problem. Therefore, a single, well-defined engineering case was selected to enable a controlled and detailed analysis. The generalisation of the proposed approach to other classes of problems is considered as a direction for future work.

2. Materials and Methods

2.1. Optimization problem formulation

We consider a part that is exposed to a multiple drop test scenario. The multiple drop test is an experimental procedure involving repeated, controlled drops of a tested object from a specified height onto a defined surface to evaluate its mechanical resistance and durability. Between successive drops, the orientation of the object is altered to reproduce various possible real-world impact scenarios (e.g., corner, edge, or flat-surface impact) (Figure 1). The number of drops, drop height, and type of impact surface are defined according to relevant standards or design specifications. This method enables the assessment of structural degradation due to repeated impact loads and, for critical components such as battery modules, allows verification of operational safety. In this example, we analyze the housing of a lithium-ion battery. Similar, as shown in the paper, approach can be used for elements and constructions under dynamic load.

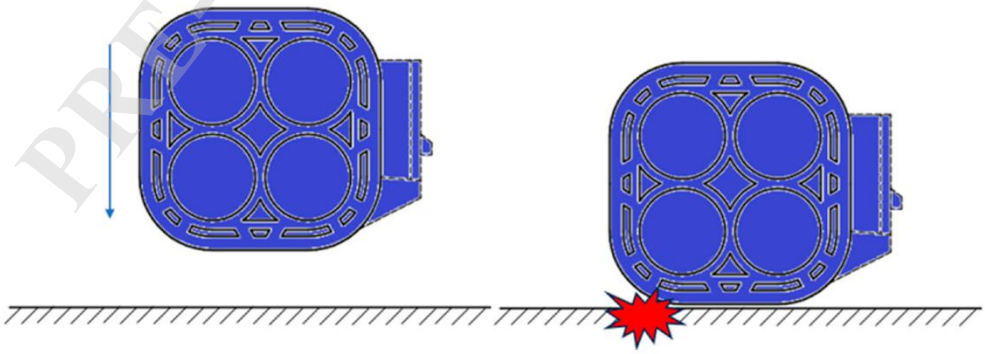


Figure 1. Object subjected to multidrop test.

The aim of the optimisation in this case is to design a structure that, when exposed to the multiple drop test, accumulate as low as possible damage effect. The following objective function was formulated:

$$\min_{P} J_0(\mathbf{P}) \tag{1}$$

we looked for the values of the design vector $\mathbf{P}$, for which the function $J_0(\mathbf{P})$ reaches its minimum value. Design vector $\mathbf{P}$ was defined as:

$$\mathbf{P} = [P_1, P_2, \dots, P_n] \tag{2}$$

and the following constraints were given:

$$\mathrm{C} = \{\mathbf{P}: \forall \mathrm{i} \in \{1, \dots, \mathrm{n}\} P_{imin} \leq P_i \leq P_{imax}\} \tag{3}$$

Where P_{imin} and P_{imax} are minimum and maximum values of i-th design variable. In the present study, the optimisation problem was intentionally formulated as damage minimisation within a fixed feasible design window. No explicit mass-related term was introduced, because the aim was to assess the behaviour of the surrogate-assisted QEA workflow for a previously bounded engineering design space. In the considered application, the housing mass itself was not treated as a critical design driver, while structural integrity under repeated drops remained the primary criterion.

Two objective (fitness) functions were used and compared. The first one, called FF1, is defined as follows:

$$J_0(\mathbf{P}) = \mathrm{V}_{del}(\mathbf{P}) + \varepsilon_{p\_max}(\mathbf{P}) \tag{4}$$

and the second, FF2:

$$J_0(\mathbf{P}) = \begin{cases} \varepsilon_{p\_max}(\mathbf{P}), & if\ \mathrm{V}_{del}(\mathbf{P}) = 0 \\ a \cdot \log(\mathrm{V}_{del}(\mathbf{P}) + 1) + b\,, & if\ \mathrm{V}_{del}(\mathbf{P}) \geq 0 \end{cases} \tag{5}$$

where a and b are coefficients are introduced later during the parameter selection step (section 3.1) for uniform distribution of fitness function values and $\mathrm{V}_{del}(\mathbf{P})$ is volume of destroyed finite elements due to damage, $\varepsilon_{p\_max}(\mathbf{P})$ is maximum equivalent plastic strain in structure during tests. FF1 defined by equation (4) is an objective function in which the optimisation goal is defined as the algebraic sum of the volume of deleted elements $\mathrm{V}_{del}(\mathbf{P})$ and maximum plastic strain $\varepsilon_{p\_max}(\mathbf{P})$. In this case, the range of the objective function values is expected to be relatively high when the number of removed due to damage elements is high and very low for case with low number of damaged elements. For this reason, it was proposed to use objective function FF2, described by (5), where a lower range of objective function values is expected due to the logarithmic function for component depending on volume of damaged elements.

The objective function values were obtained from an explicit FEM model of the battery housing subjected to the prescribed sequence of drop events. The model was parameterised by the

design vector $\mathbf{P}$, and for each candidate design the same multi-drop loading scenario was simulated. Each objective-function evaluation corresponded to the full prescribed drop sequence for a given design. The structural state after each impact was transferred to the subsequent drop, and the reported values were extracted after completion of the entire sequence. The quantities extracted from the FEM analysis were the maximum equivalent plastic strain $\varepsilon_{p_\max}(\mathbf{P})$and the deleted-element volume $V_{del}(\mathbf{P})$, which were then used to calculate the fitness functions FF1 and FF2. Damage was represented by damage initiation followed by damage evolution with stiffness degradation and element deletion. The FEM calculations were embedded in an automated script that evaluated consecutive design vectors and collected the quantities required for the objective function, as shown in Figure 2.

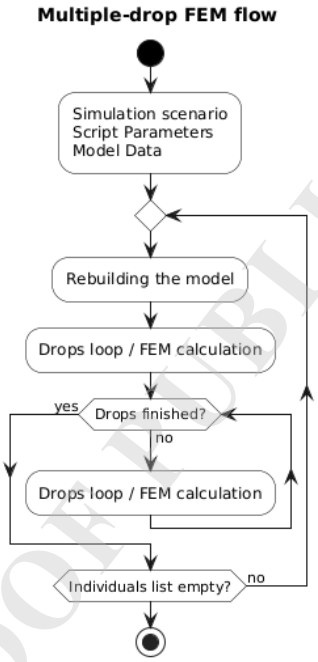


Figure 2. Scheme for obtaining objective function value results from FEM.

Cumulative damage is defined as a combination of the plastic strain calculated in Abaqus $\varepsilon_{p\_max}(\mathbf{P})$ and the volume of deleted elements $\mathrm{V}_{del}(\mathbf{P})$. The deleted volume is calculated on the base of sum of all removed elements (when regular mesh is used). During FEM analysis, elements are removed from the model on the base of the damage initiation model [17] and the constitutive model of damage evolution. After the onset of damage, determined on the damage initiation model, the damage evolution process begins to reduce the stiffness of the element. Once the element loses stiffness, it is removed from the further steps of simulation.

2.2 Database of solutions and design of experiment

The results of multiple analyses mentioned above are collected in database. Each row contains the values of the vector $\mathbf{P}$ **,** and the following columns contain analysis results such as the number of deleted elements, the volume of deleted elements, and the maximum equivalent plastic strain. For comparison purposes, two types of databases were evaluated. The first one

is based on design vector obtained with use of design of experiments (DOE) techniques. In the case all designs vectors were generated using Latin Hypercube Sampling (LHS) [18] with the CenteredMaxMin option. The LHS method with the centered-maximin criterion combines two strategies to improve the quality of sample distribution. Each sample is placed at the center of its stratified interval (centered), ensuring uniform coverage along individual dimensions. Simultaneously, the configuration is optimized to maximize the minimum Euclidean distance between any pair of points (maximin), which improves the overall space-filling properties. This approach results in well-distributed, low-correlation sampling designs, which are particularly suitable for surrogate modelling and design of experiments in high-dimensional spaces [19].

The few databases were created. The full, largest database was generated and several smaller databases were created to compare prediction quality of metamodels. The design vectors in all databases were generated separately. To avoid high computational costs, points from each smaller databases corresponding to the nearest points in the larger dataset were identified. These selected points were then transferred to a new, smaller dataset, as shown in Figure 3.

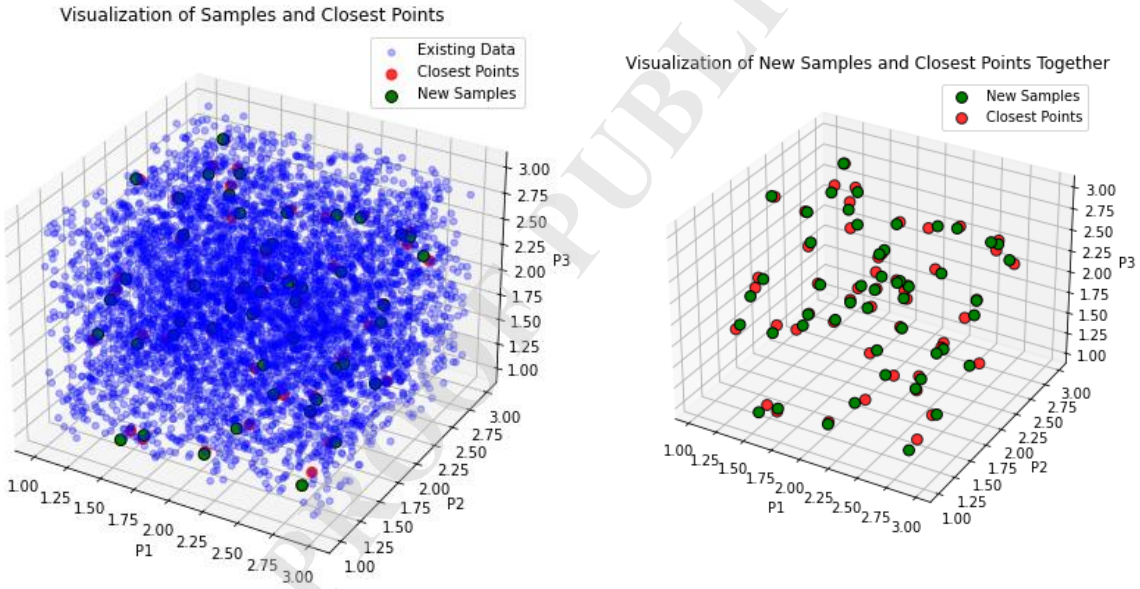


Figure 3. Selecting the closest points from the smaller database into the full database.

It is important to note that the distance between points does not have to be calculated in real (three-dimensional) space. The distance can be computed using (6), even in a higher-dimensional space beyond the physical domain. This was done by applying the Euclidean distance between points in an n-dimensional space.

$$d(a,b) = \sqrt{\sum_{i=1}^{m}(a_i - b_i)^2} \tag{6}$$

where a_i and b_i is value of variables for two different points and m is number od dimension. This approach enabled a significant reduction of computational time while maintaining a uniform distribution of the samples.

Another database was based on design vectors selected from previous runs of optimisation using QEA [16]. Previously collected data from multiple runs of the optimisation task provided a sufficient amount of data. The data in this database are not as evenly distributed as in the case of the LHS method. However, the advantage of this approach is that the database contains individuals that may be close to the global minimum.

2.3 Metamodels for objective function prediction

The determination coefficient R^2 was selected for assessing the prediction efficiency. Although it does not guarantee that the model with the highest R^2 provides the most accurate prediction, this metric is useful to compare different metamodels [20–23]. When combined with validation loss and training loss, it offers a good indication of whether the function is well-fitted to the data. Hyperparameter tuning tools for ANN, such as Optuna [24], help avoid overfitting by stopping the training process when the validation loss becomes greater than the training loss. This approach serves as a reliable indicator. Prediction efficiency and the R^2 coefficient depend heavily on the input data. It is well known that data must be carefully prepared before being used in metamodeling. Several preprocessing techniques are applied to improve the quality of input data, such as normalisation, scaling, and outlier removal. Metamodels generally perform better when the input data are more uniformly distributed. In this case, the dataset showed a tendency to produce a high number of results without any damage, which led to a non-uniform distribution in the results database. Optimisation based on a model trained on such data may result in high prediction errors. In this study, the primary focus is on the design space where no damage is observed. Therefore, it was essential to allocate more attention and achieve higher prediction accuracy within this specific region [25, 26]. Two types of databases were used to store the data: the first one (full) containing the full range of objective function values, while the second database (no damage) included only design vectors for which no damage occurred (no elements were deleted during FEM simulations). The model selection logic is presented in Figure 4.

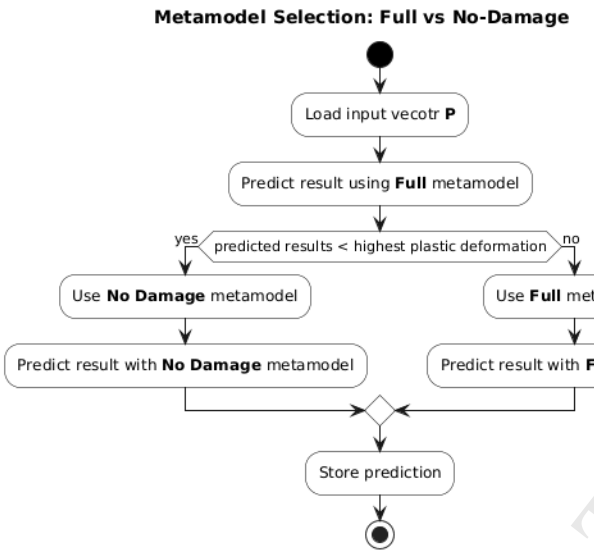


Figure 4. Model selection logic.

When the predicted damage exceeds the threshold of the maximum possible plastic strain, the metamodel returns a prediction based on the model trained on the full database. Otherwise, if the value of the fitness function indicates that no damage occurred in the model, the prediction is made using the database that contains only undamaged individuals. It allows to obtain good predictions in design space which is the most interesting prom point of view of this problem, and obtain good overall view, while using Full database.

2.4 Metamodels and hyperparameters tuning

The ANN, Kriging method and Polynomial Regression were used in the paper. As artificial neural networks (ANNs) have become more widespread in various industrial fields, new tools for hyperparameter optimisation have been developed in recent years [27]. The popular ANN hyperparameter tuning methods are Grid Search and Random Search [28]. The Grid Search method allows for an exhaustive search through all configurations, but it is computationally inefficient. The Random Search method is less accurate, as it carries the risk of missing the optimal configuration but has lower computational costs. ANN has a large number of hyperparameters that must be tuned. The following activation functions were considered : ReLU, tanh, sigmoid, softmax, softplus, softsign, SELU, ELU, exponential, linear, and swish. Regularisation were applied to the dataset, along with techniques such as neuron dropout. The dropout rate was treated as a tunable parameter that controls the fraction of neurons randomly "dropped out" (set to zero) during each training iteration. Next, the number of hidden layers can be defined. Once the number of layers is set, it is important to tune the number of neurons in each hidden layer. The objective is to achieve the highest possible R^2 value while minimizing the training time.

To construct the Kriging metamodel, it was essential to select the type of variogram function (*variogram model*). This function describes how the variance changes with the spatial distance

between samples. Five standard models were evaluated: linear, power, Gaussian, spherical, and exponential. In cases involving a small number of variables, a Grid Search approach was used to explore all possible configurations [29].

When constructing a metamodel based on Polynomial Regression (PNR), a key aspect of hyperparameter tuning is selecting the appropriate order (degree) of the polynomial. A polynomial degree that is too low may result in underfitting, where the model fails to capture important trends in the data [30, 31]. On the contrary, an excessively high degree increases the risk of overfitting and introduces unnecessary computational complexity. In practice, the optimal degree is typically determined using cross-validation or a dedicated validation dataset. Popular hyperparameter tuning methods for polynomial regression include Grid Search and Random Search. However, when faced with a large number of possible configurations, more advanced strategies, such as Bayesian optimisation, can be advantageous, as they allow for a more efficient identification of promising parameter sets. In practice, the tuning process involves iteratively testing models with different polynomial degrees and regularization parameters, and then selecting the solution that provides the best results (for example, the highest coefficient of determination R^2 with the lowest validation error). Due to quite small database base and small number of variables, Gridsearch technique was used, after checking every degree of PNR. To determine the most reliable polynomial regression model, a custom ranking criterion was introduced instead of simply selecting the configuration with the highest test R^2. For each trained model (defined by a specific polynomial degree and data split), three performance metrics were computed: R^2 on the training set (R^2_train), validation set (R^2_val), and test set (R^2_test). The final selection was based on a composite score, defined as the mean of the three R^2 values, penalized by their pairwise absolute differences. Mathematically, the score for each model was computed as follows:

$$\text{score} = \frac{R^2_{\text{train}} + R^2_{\text{val}} + R^2_{\text{test}}}{3} - |R^2_{\text{train}} - R^2_{\text{val}}| - |R^2_{\text{val}} - R^2_{\text{test}}| \quad (7)$$

This approach prioritizes models that not only achieve high predictive performance, but also maintain consistency across all data subsets, reducing the risk of overfitting or poor generalization. The configuration with the highest score was then selected as the final model for each dataset variant (i.e., full and no-damage). This ensures that the chosen polynomial degree is both accurate and robust with respect to unseen data.

2.5 QEA optimisation with metamodels

The QEA used in this study is the previously published algorithm described in [16]; the present work focuses on metamodel selection, hyperparameter tuning, and the evaluation of hybrid strategies for embedding surrogate predictions into that workflow. Running optimisation based only on previously generated databases and static metamodel predictions may not reflect a viable engineering solution. However, such approach was used for testing purposes. Solving the

task exclusively using QEA and evaluating the fitness function with finite element method (FEM) calculations leads to excessively long computational times. Previous studies have shown that a single optimisation run, without metamodels, may require up to 40 hours, and repeating the process at least four times results in a total computation time of approximately 150 hours (sometimes it should be repeated more times but the time consumption is too high). To reduce the computational cost, a hybrid approach is proposed. In the context of the previously developed QEA workflow, a practical extension is to generate the database dynamically during the optimisation process and use a metamodel to accelerate objective (fitness) function evaluation.

Two strategies are proposed, each with two phases. In phase A, the objective (fitness) function is calculated only using FEM. In phase B, the process is supported by metamodel predictions based on the data collected in the first phase. This flowchart of the algorithm is shown in Figure 5.

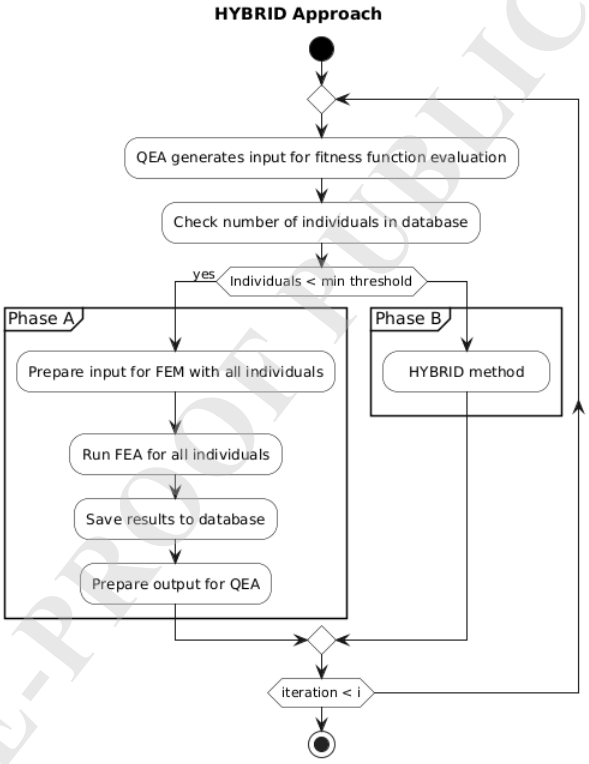


Figure 5. Flowchart of the HYBRID algorithm.

Two different HYBRID approaches will be tested. The first approach, called HYBRID1, is based on the logic that, in each iteration of phase B, all objective (fitness) function evaluations are performed using the metamodel. The best result is then recalculated using FEM. The obtained value is used to update database. The ANN in next steps is trained including new solutions. Scheme in Figure 6 shows workflow of HYBRID 1.

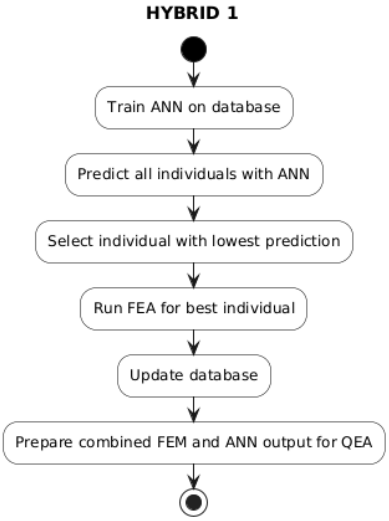


Figure 6. Scheme of HYBRID 1approach in the phase B.

In contrary to HYBRID1,in the HYBRID2 approach, obligatory FEM calculations were omitted in every optimisation iteration. If the best individual in a generation had already been evaluated using FEM, its result was retrieved from the database. With each new step, fewer predictions were saved, until eventually none were written. Furthermore, the network was not retrained in every iteration, just in case where at least two new individuals were added to database. If no new individuals were introduced, the prediction model remained unchanged. This approach

allows to obtain more steady results, but in case of new entries, metamodel is recreated again. This approach is presented in Figure 7.

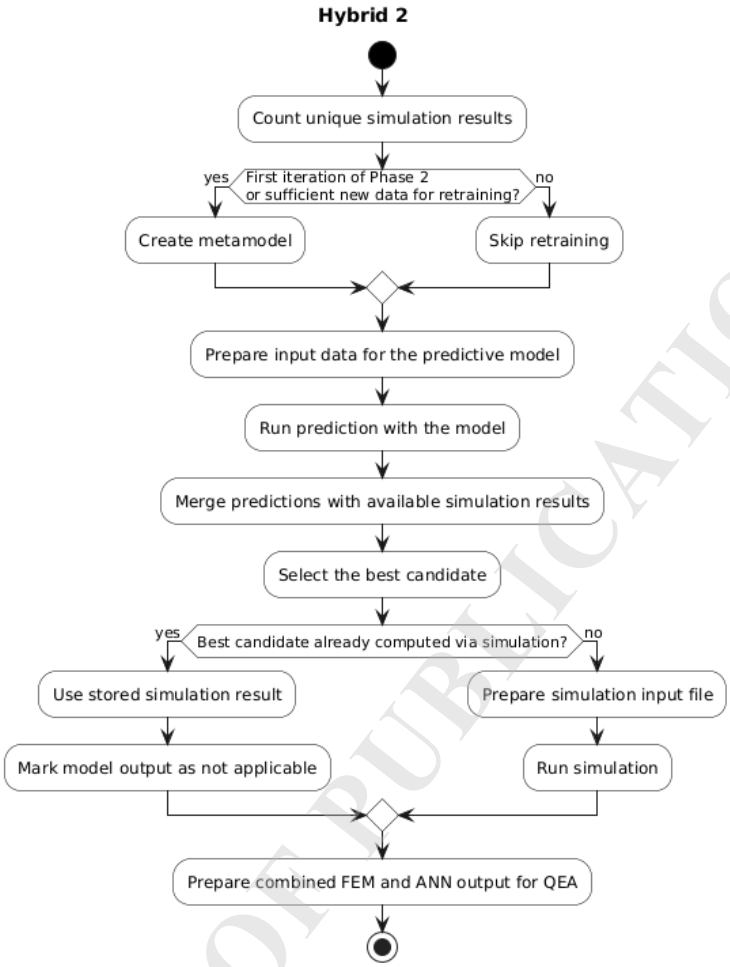
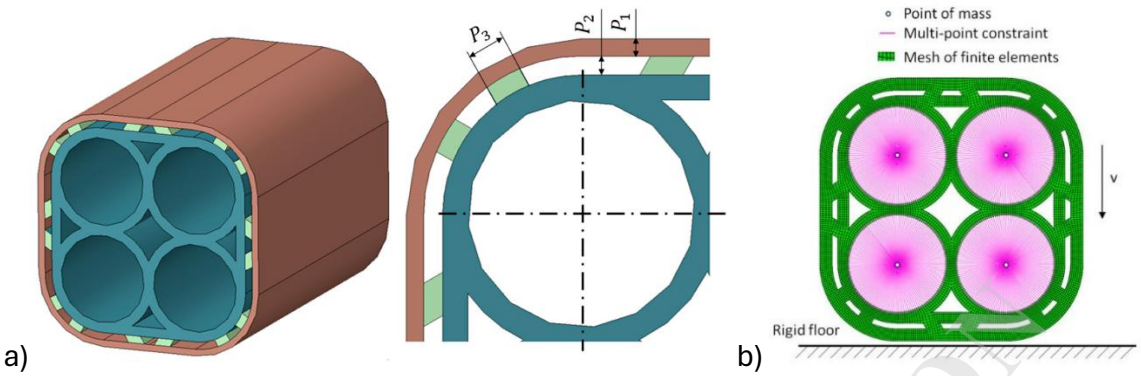


Figure 7. Scheme of the HYBRID2 approach in the Phase B.

3. Numerical example description

To evaluate the approach proposed in this article, a conceptual battery design was created, defined by a design parameters vector **P** consisting of three parameters. The values of parameters are constrained in the range from 1mm up to 3mm. This range was selected based on practical wall-thickness limits for the considered injection-moulded housing concept. In the analysed application, the housing mass itself was not treated as the primary design driver, while manufacturability and impact resistance were treated as the main constraints. The geometry of the structure is shown in Figure 8a. This case was intentionally selected as a representative and computationally demanding problem, allowing for systematic comparison of different metamodeling techniques, datasets, and optimisation strategies within a consistent framework.



a) b)

Figure 8. a) Geometry of the battery housing, b) the cells point-mass locations.

On the basis of this design, a finite element method (FEM) numerical model was developed. The housing was discretized with approximately 64,000 eight-node linear solid elements, using an average element size of 0.4 mm and at least four elements through the wall thickness. The battery cells were represented by 0.07 kg inertia points coupled to the housing. The inertia points were connected to the housing by a multi-node constraint, while all degrees of freedom of the analytical rigid surface were fixed. The drop event was modelled as impact against a fully fixed analytical rigid surface, with the initial velocity prescribed normal to that surface. The enclosure material was represented by a homogenized isotropic PA6-GF30 model with Young's modulus of 9.1 GPa, Poisson's ratio of 0.34, and density of 1350 kg/m^3, combined with von Mises plasticity, bilinear hardening, and a triaxiality-dependent ductile damage initiation model. The highest plastic equivalent strain value in this material was determined as 1.014. The metamodels "full" and "no damage" uses this threshold as the switching criterion. The drop sequence included two drops in the same direction, corresponding to the two free-fall velocities: 10.3 m/s and 16.5 m/s. Further implementation details are given in [16].

The hyperparameters of the metamodels were trained on the base of the FEM results obtained for above problem. After performing calculations for 5000 samples generated using LHS, the *LHS5000* database was created. The distribution of the results in database is shown in Figure 9, where P1 ,P2 and P3 are parameters and Del_El_Volume is $\mathrm{V}_{del}(\mathbf{P})$ and PEEQ is $\varepsilon_{p\_max}(\mathbf{P})$.

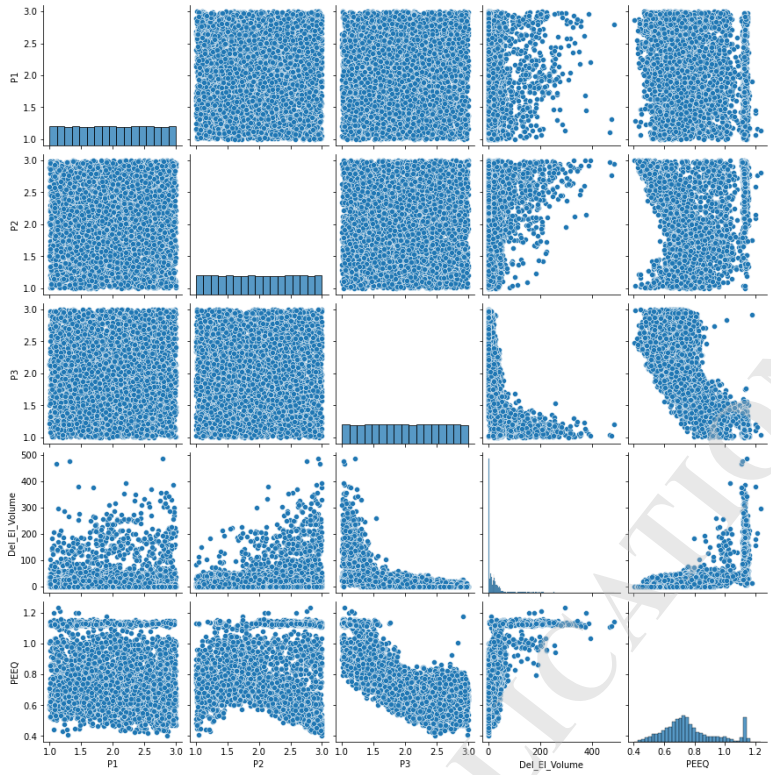


Figure 9. Review of the LHS database.

As was mentioned in previous chapter, also smaller databases was crated *LHS2500, LHS1225, LHS 625, LHS312, LHS156* with 2500,1225,625,312 and156 samples.

The data obtained from multiple runs of the optimization task described in [16] were merged into a single database. The combined dataset was then randomly sorted and divided into smaller subsets. The QEA based databases included the following subsets: *QEA1687, QEA625, QEA312*, and *QEA156*. The largest database (*QEA1687*) is shown in Figure 10.

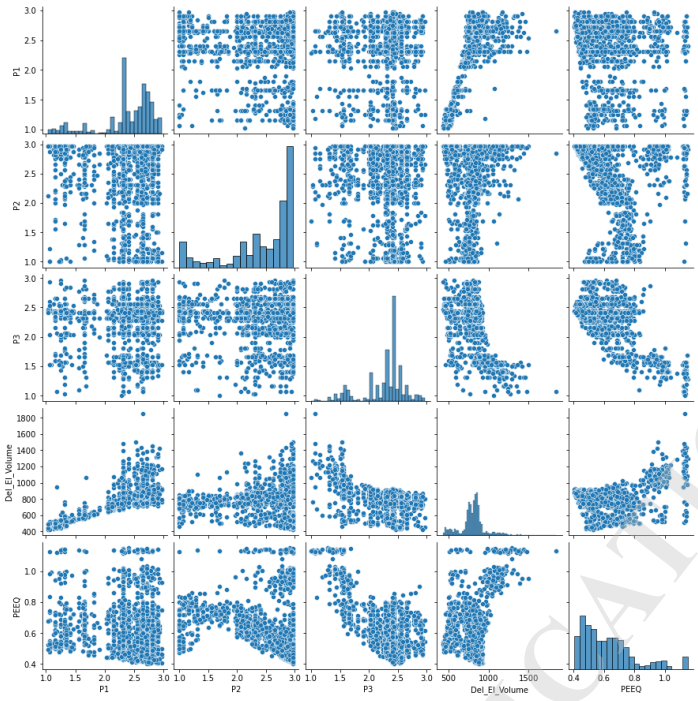


Figure 10. Review of the *QEA1687* database.

3.1 ANN tuning of hyperparameters.

To establish a universal configuration applicable to all databases, the *LHS5000* database was selected due to its uniform distribution and the largest number of samples, providing a solid foundation for hyperparameter optimisation. For hyperparameter tuning, objective function FF1 (4) was used. Regularization was applied during HPO, and a dropout rate of 20% was used to randomly deactivate neurons in order to prevent overfitting. The Adam [32] optimisation algorithm was used. The initial configuration consisted of a two-layer artificial neural network (ANN) with 30 neurons in each hidden layer. The maximum number of training epochs was set at 1000, with an early stopping mechanism triggered if no improvement in validation loss was observed within 50 consecutive epochs. The data were standardized using a standard scaler and divided into three sets: training data – 80%, validation data – 10%, and test data – 10%. The optimisation process included 500 trials. Since Optuna adaptively allocates trials to promising regions of the search space, the number of evaluations assigned to individual activation functions was not balanced. Therefore, the results reported in Table 1 should be interpreted as a descriptive summary of the hyperparameter search process rather than as a controlled comparison with equal evaluation budgets for each activation function. The results of this study are presented in Figure 11 and Table 1.

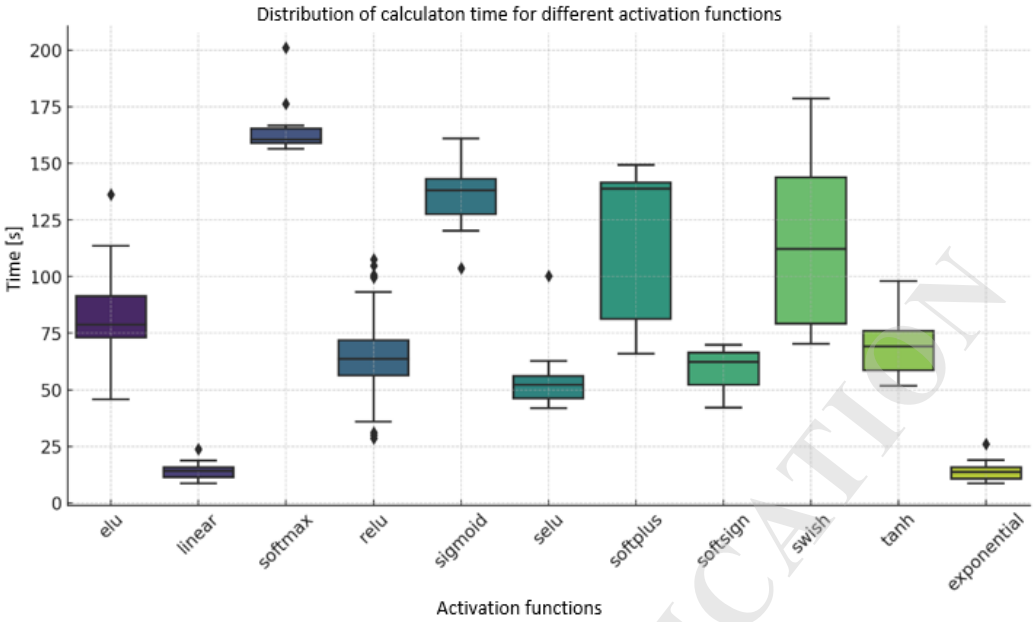


Figure 11. Calculation time for various activation functions.

Table 1. Descriptive summary of activation-function results obtained during the Optuna search.

Activation Function	Trials	R^2	std. dev. R^2	Average time [s]
ReLU	347	0.8155	0.0032	64.46
Sigmoid	15	0.8135	0.0022	135.32
Swish	15	0.8108	0.0053	113.73
Softplus	15	0.8014	0.0063	119.53
ELU	16	0.7924	0.0063	82.22
Tanh	16	0.7821	0.0071	109.87
Leaky ReLU	16	0.7758	0.0085	91.34
Softmax	15	0.6212	0.0124	159.34
Linear	15	0.3628	0.0093	18.69
Hard Sigmoid	15	0.3415	0.0152	22.41
Exponential	15	0.2894	0.0185	31.57

In the performed Optuna search, ReLU was associated with the highest average R^2, the lowest standard deviation, and a relatively short computation time among the frequently selected configurations. However, because the search budget was allocated adaptively and unequally

across activation functions, these results should be treated as a descriptive outcome of the hyperparameter optimisation procedure rather than as a strictly controlled comparison. For this reason, ReLU was adopted in the subsequent experiments as the most promising activation function identified within the performed search. Using the activation function identified in the previous study, a new experiment was conducted to investigate the simultaneous effect of the number of layers (ranging from 1 to 4) and the number of neurons (ranging from 2 to 100). A total of 100 tests were performed. The results are presented in the following graph. The findings indicate that these hyperparameters have a minimal impact on both prediction accuracy and computational time. Networks with a single hidden layer required significantly more computation time. Networks with three or four layers performed faster and delivered better results, but were less stable and occasionally produced low quality predictions, as illustrated in Figure 12.

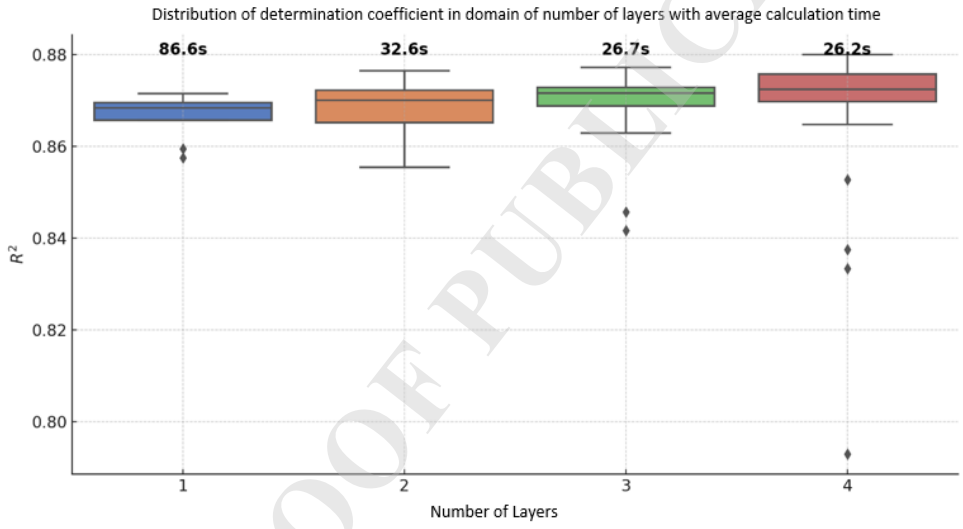


Figure 12. Results from various hidden layers neurons calculation time.

Since there was no significant difference in average training time and the model remained stable, subsequent tests used networks with two hidden layers. To determine the optimal number of neurons in each layer, the grid search method was applied. This method evaluates all possible combinations and enables the visualization of R^2 values using a heat map. The results are shown in Figure 13.

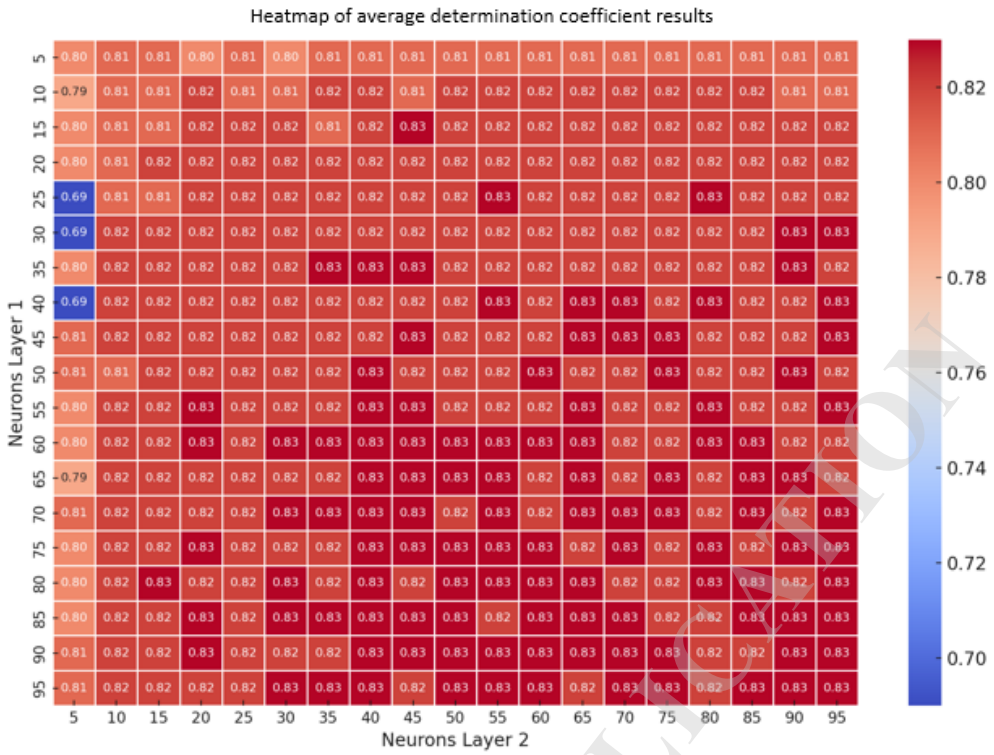


Figure 13. Heat map of the coefficient of determination for various quantity of neurons in layers.

To evaluate the impact of different validation sets on training time and model prediction accuracy, an experiment was conducted using five distinct initialization of the validation, test, and training datasets. These datasets remained consistent within each individual test, but varied between repetitions within the same test configuration. In total, 1800 neural network training sessions were performed. The results are illustrated in Figure 13 and Figure 14.

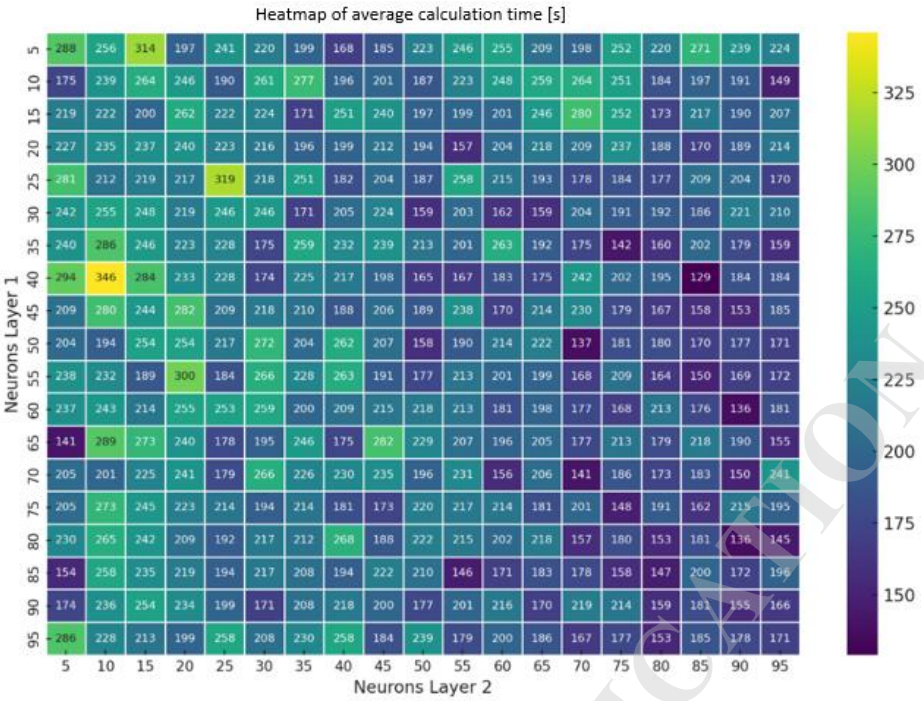


Figure 14. Heat map of time calculation for various number of neurons in layers.

For the subsequent tests, it was decided that a configuration with at least 50 neurons in each layer would ensure stability and maintain a short computation time. An example of learning curves for FF1 is shown in Figure 15.

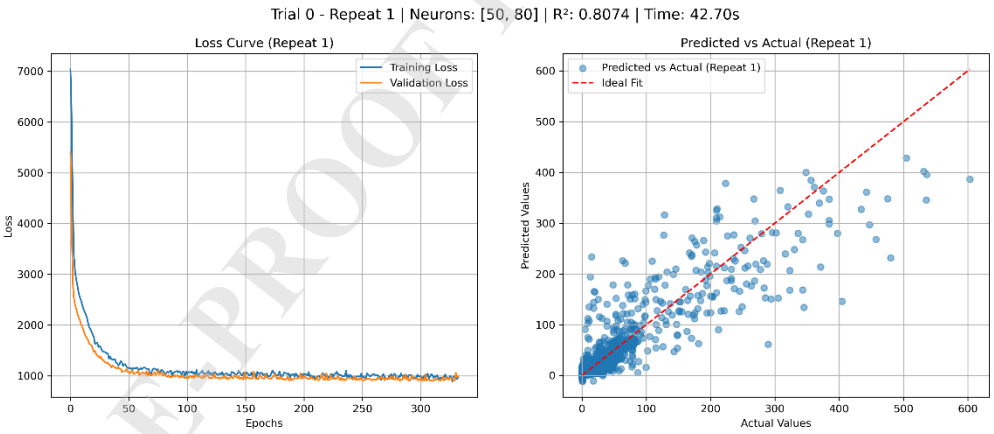


Figure 15. Learning curves for the **Fitness Function 1.**

The parameters of objective function FF2 (5) were adjusted by setting $a = 0.3194$ and $b = 1.014$, so that the function equals 1.014 at zero volume, which corresponds to the maximum plastic deformation $\varepsilon_{p\_max}(\mathbf{P})$ value which can be obtained for this kind of material. At the

maximum volume of approximately 500 mm^3, the function approaches a value of about 3.

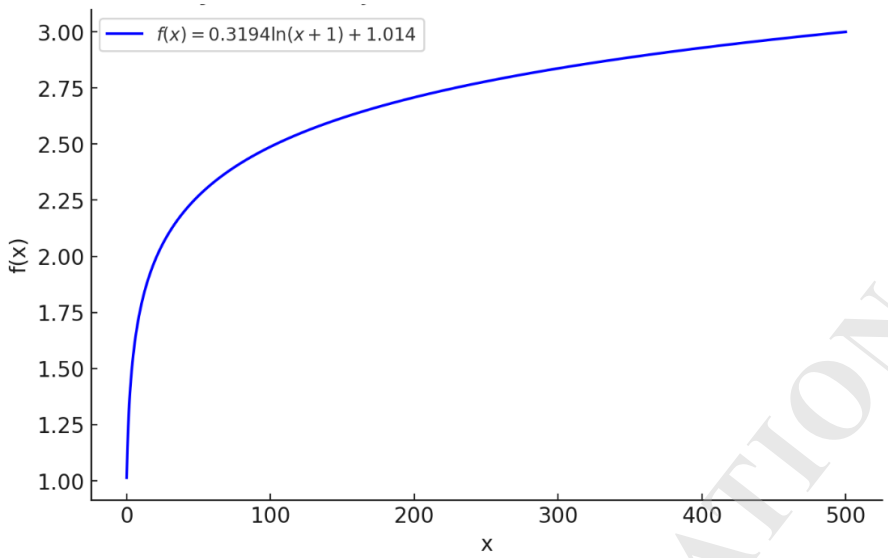


Figure 16. Selection of parameters for FF2, to obtain a compact distribution of results.

Predicted vs actual Values graph shows that the values have a smaller spread what is visible in Figure 17.

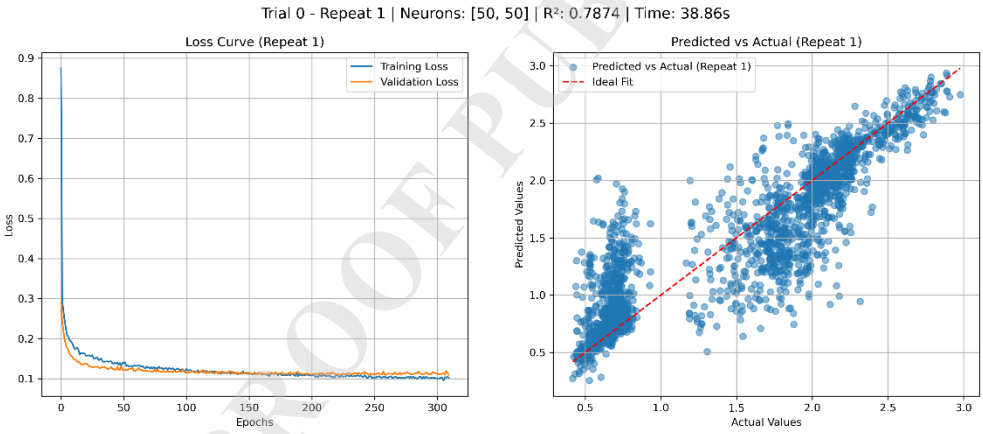


Figure 17. Learning curves of the Fitness Function2.

Both learning curves show that it exist problematic area between results with damage and no damage to prediction. Therefore it was decided to use separate ANN for prediction from “full” model and “no damage” database.

Results of training with separated ANN model on both “full” and “no damage” with comparison for FF1 and FF2 is shown in Figure 18.

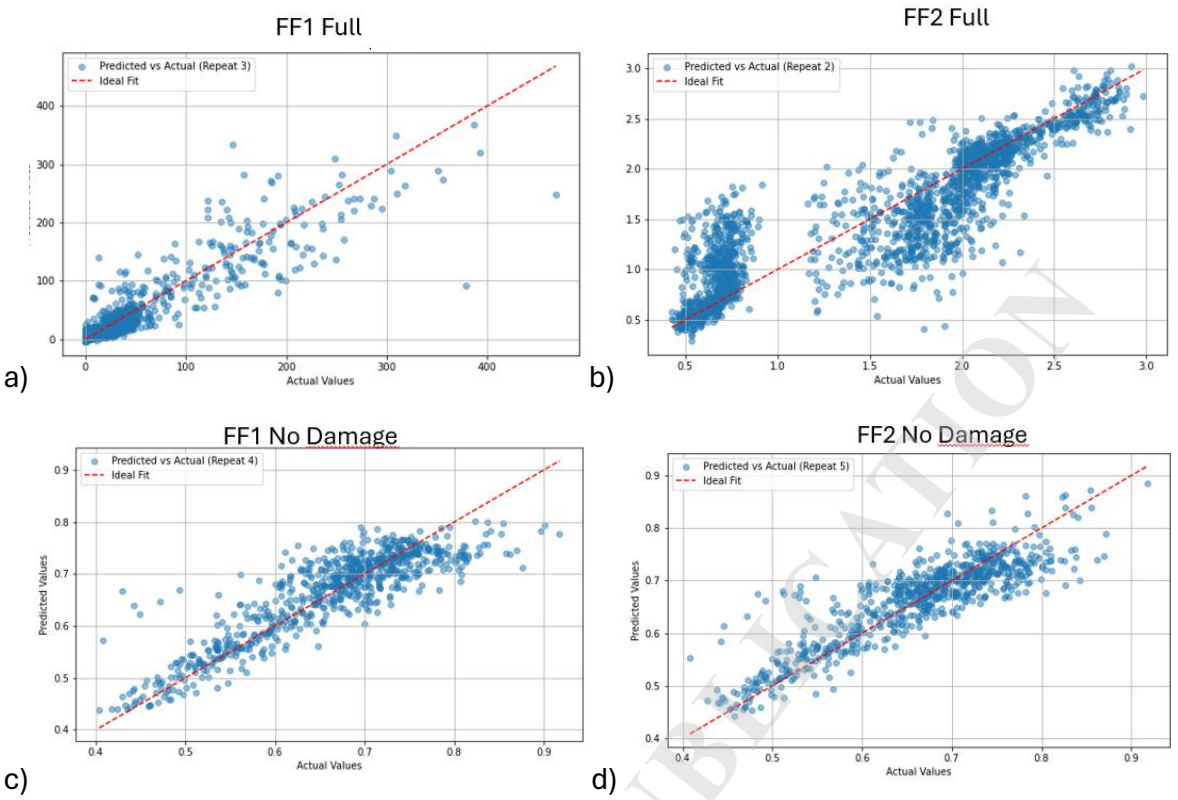


Figure 18. An example of the learning curves of "full" (a) and b)) and "no damage" models (c) and d)).

The next step was to determine whether different datasets, such as QEA and smaller LHS subsets, would yield similar and good results. The following table presents a set of results obtained from training the artificial neural network (ANN) five times on each dataset. The average values were recorded in Table 2.

Table 2. R^2 for different Fitness Functions and sizes of databases.

ANN with FF1 based on database LHS						
Database	R^2		Val Loss		Train Loss	
LHS	Full	No damage	Full	No damage	Full	No damage
LHS5000	0.8311542	0.7491974	426.7850708	0.0022338	452.4532778	0.0021394
LHS2500	0.824483	0.765545	445.712219	0.002176	377.840295	0.002113
LHS1250	0.837276	0.740888	455.713214	0.002262	527.78225	0.002236
LHS625	0.757262	0.568795	813.327331	0.003634	1092.683227	0.004399
LHS312	0.826518	0.625481	583.259711	0.003101	660.375433	0.005706
LHS156	0.635194	-0.607225	731.145435	0.011335	1519.446338	0.013695
ANN with FF2 based on database LHS						
Database	R^2		Val Loss		Train Loss	
LHS	Full	No damage	Full	No damage	Full	No damage
LHS5000	0.779647	0.756274	0.113400	0.002171	0.109448	0.002031

LHS2500	0.753563	0.763015	0.127057	0.002203	0.128848	0.002258
LHS1250	0.733327	0.729156	0.135273	0.002366	0.140219	0.002363
LHS 625	0.705947	0.587907	0.157150	0.003478	0.170239	0.004043
LHS 312	0.733171	0.328765	0.141381	0.005361	0.158526	0.009317
LHS156	0.641887	-1.0251	0.173655	0.013695	0.141143	0.016411
ANN with FF1 based on database QEA						
Database	R^2		Val Loss		Train Loss	
QEA	Full	No damage	Full	No damage	Full	No damage
QEA1687	0.748013	0.832073	197.507169	0.001692	255.213092	0.002015
QEA625	0.517468	0.847087	285.901752	0.001646	139.069743	0.001530
QEA312	0.519403	0.866713	126.929550	0.001416	166.399509	0.001367
QEA156	0.327259	0.460945	70.565874	0.006220	96.116487	0.006020
ANN with FF2 based on database QEA						
Database	R^2		Val Loss		Train Loss	
QEA	Full	No damage	Full	No damage	Full	No damage
QEA1687	0.750640	0.860533	0.115640	0.001528	0.117122	0.001543
QEA625	0.756393	0.830961	0.115566	0.001924	0.133548	0.002240
QEA312	0.663904	0.776071	0.163220	0.002354	0.181552	0.002585
QEA156	0.629303	0.245422	0.206355	0.008436	0.131287	0.007253

The differences between the fitness functions appear to be relatively small; however, FF2 demonstrated higher R^2values in the smaller datasets, which is particularly noteworthy in this context. Therefore, subsequent tests were conducted using FF2.

After the hyperparameter tuning process was completed, the next step was to apply this knowledge to assess prediction performance. The study involved two primary computational tasks:

- optimisation using the full input set, which consisted of 5,000 individuals generated via Latin Hypercube Sampling (LHS), and,

- evaluation of the prediction model's performance using only the data obtained from the QEA process, which included 1,687 individuals.

Table 3 presents the results of 16 separate optimisation tests. These tests were carried out for four different prediction models. Each model was checked four times using the QEA algorithm. The aim of the analysis was to examine the variability of the results across repetitions of the same model.

Table 3. ANN Prediction vs FEM results.

QEA FF2 – based on *LHS5000*				
Model [R^2]	Trial	Result	Individual	FEM Result
Full= 0.792	1	0.439186	2.375,2.9375,2.4375	0.44241
	2	0.435265	2.8125,2.9375,2.625	0.44878
ND=0.78	3			

	4	0.439186	2.375,2.9375,2.4375	0.44241
Full= 0.7847 ND=0.771	1 2 3 4	0.461702	2.9375,2.9375,2.5	0.46484
Full=0.7844 NDD=0.754	1 2 3 4	0.45215	2.25,2.9375,2.5	0.445811
Full=0.7820 ND=0.751	1 2 3 4	0.431665	2.9375,2.9375,2.5	0.46484
QEA FF2 based on *QEA1687*				
F=0.7213 ND=0.8713	1 2 3 4	0.437105	2.75,2.9375,2.4375	0.41423
F=0.7140 ND=0.8761	1 2	0.439528	2.6875,2.9375,2.4375	0.411295
	3	0.439374	2.5625,2.9375,2.5	0.42695
	4	0.438613	2.625,2.9375,2.4375	0.42717
F=0.7260 ND=0.8721	1 2 3 4	0.438376	2.75,2.9375,2.5	0.434021
F=0.7396 ND=0.9749	1 2 3 4	0.448706	2.6875,2.9375,2.5	0.42717

Figure 19 shows the fitness function value in following iterations.

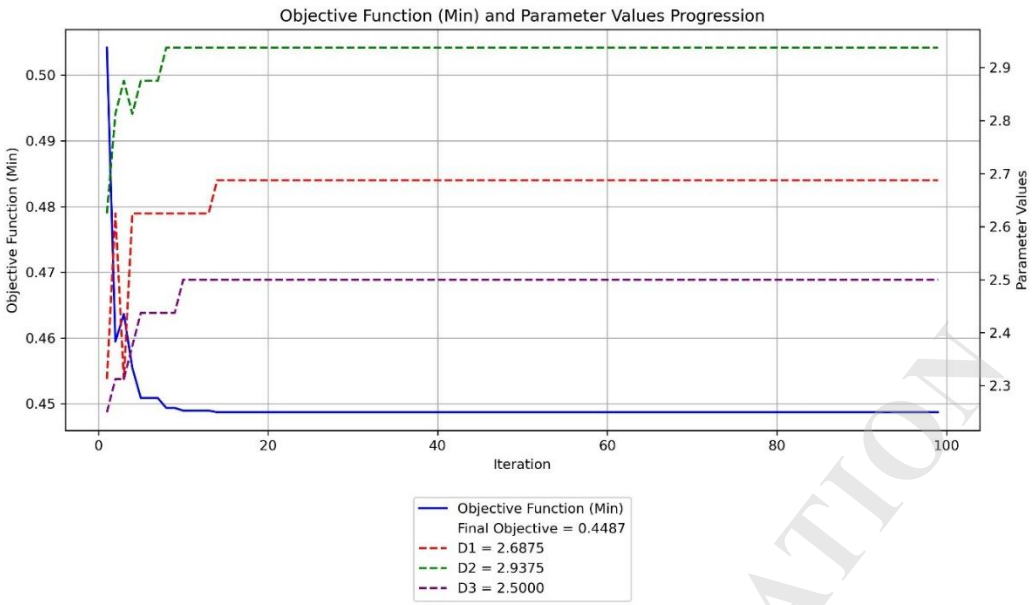


Figure 19. Fitness function value in function of iteration number.

3.2 Kriging based metamodel

Each variogram model - linear, power, Gaussian, spherical, and exponential - was evaluated five times using different training and validation dataset splits. Hyperparameter tuning was performed using the Optuna library in GridSampler mode, which produced the average prediction quality (R^2) for each configuration. The variogram model that achieved the best fit (linear) was selected for the final metamodel, as shown in Figure 20.

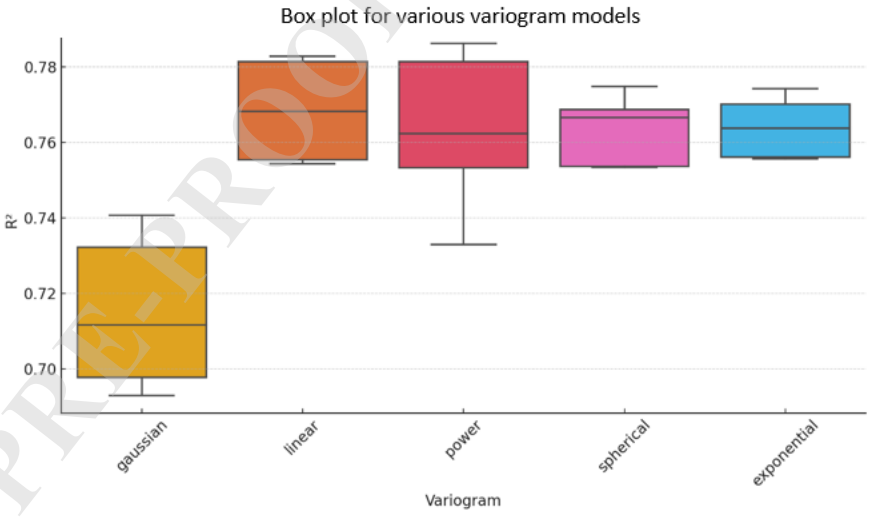


Figure 20. HPO results with model of variogram for kriging.

Table 4 shows the results of the use of kriging in both LHS and QEA databases in different sizes of the database.

Table 4. R^2 in several size of databases LHS and QEA databases.

Kriging z FF2 based on databases LHS						
Database	R^2		Val Loss		Train Loss	
LHS	Full	No damage	Full	No damage	Full	No damage
LHS5000	0.782909	0.756464	0.214571	0.030546	0.334160	0.046494
LHS2500	0.754501	0.761495	0.234398	0.030487	0.355564	0.046969
LHS1250	0.760380	0.773227	0.241413	0.030536	0.348399	0.044433
LHS625	0.766255	0.538562	0.248900	0.045064	0.353607	0.061690
LHS312	0.734848	0.618931	0.258842	0.041498	0.374150	0.055653
LHS156	0.660231	0.228298	0.307563	0.057847	0.405318	0.072067
Kriging with FF2 based on databases QEA						
Database	R^2		Val Loss		Train Loss	
QEA	Full	No damage	Full	No damage	Full	No damage
QEA1687	0.773371	0.858205	0.202413	0.025761	0.323797	0.039368
QEA625	0.768539	0.816137	0.213611	0.033606	0.330643	0.045771
QEA312	0.737385	0.772807	0.242555	0.033486	0.357599	0.048491
QEA156	0.674119	0.656937	0.298575	0.041870	0.424546	0.061597

The QEA tasks relied exclusively on predictions from the Kriging metamodel. Four different models were tested, each trained and evaluated on a distinct dataset. The same model was not tested multiple times because previous experiments indicated that the results were highly consistent. The results are presented in Table 5.

Table 5. Kriging prediction vs. FEM.

QEA FF2 – based on *LHS5000*			
Model [R^2]	Result	Individual	FEM Result
F-0.7954 ND – 0.8034	0.43023	2.6875,2.9375,2.5	0.42717
F-0.7795 ND –0. 7269	0.428881	2.8125,2.9375,2.4375	0.42132
F-0.7886 ND – 0.7942	0.423571	2.625,2.9375,2.5	0.42642
F-0.7885 ND – 0.78	0.43848	2.8125,2.9375,2.4375	0.42132
QEA FF2 – based on *QEA1687*			
Model [R^2]	Result	Individual	FEM Result
F- 0.7993 ND- 0.8657	0.427507	2.6875,2.9375,2.5	0.42717
F- 0.8023 ND- 0.870	0.425319	2.6875,2.9375,2.5	0.42717

F- 0.7857 ND- 0.8735	0.422793	2.75,2.9375,2.5	0.43402
F- 0.7957 ND- 0.8643	0.420768	2.75,2.9375,2.5	0.43402

On average, the Kriging method produced better results. However, the surrogate neural networks (ANN) identified individuals with lower values of objective functions. In general, the performance of both methods was similar.

3.3 Polynomial Regression based metamodel

Polynomial Regression tuning of hyperparameters showed tendency, that for largest *LHS5000* database, during degree increasing, in lower range of polynomial degree, the model are not well fitted to data. Later it achieve maximum value of $R^2$and R^2on validation set, and with higher degree model seems to be overfitted, and R^2usually is below 0. Figure 21 shows the distribution of the determination coefficient of the training set and the validation set. Model with R^2below zero was omitted in visualization.

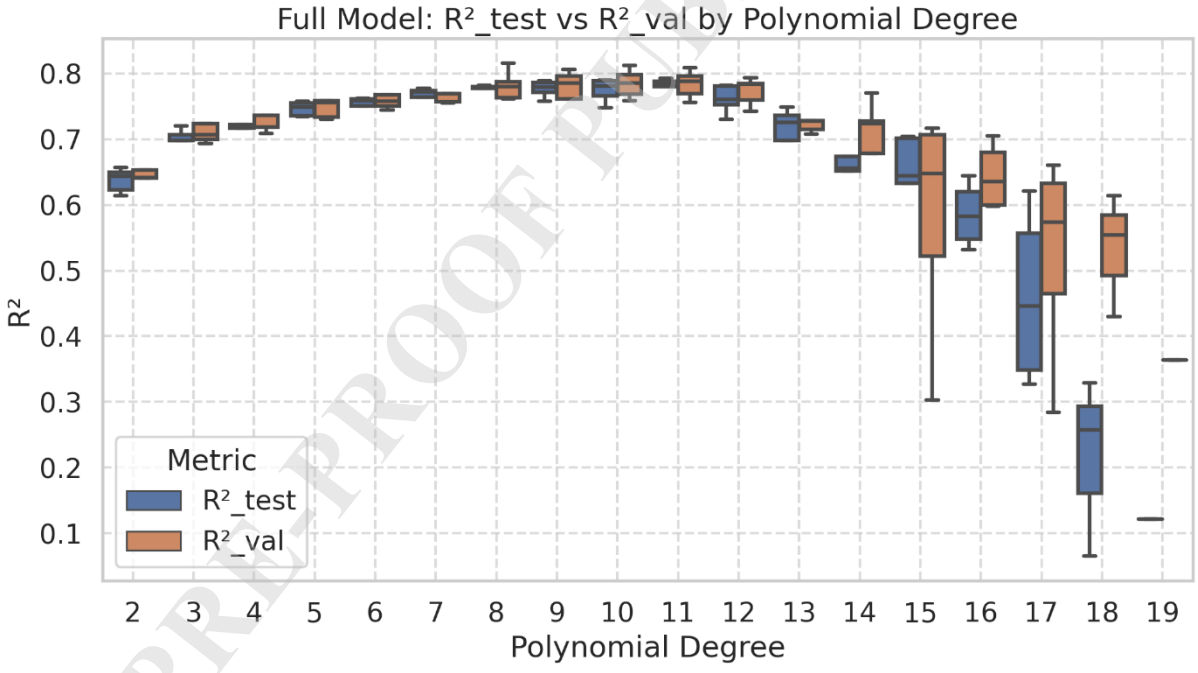


Figure 21. Polynomial Regression degree selection for *LHS5000* database in Full Model.

The same situation happened for the “no damage” model that is shown in Figure 22.

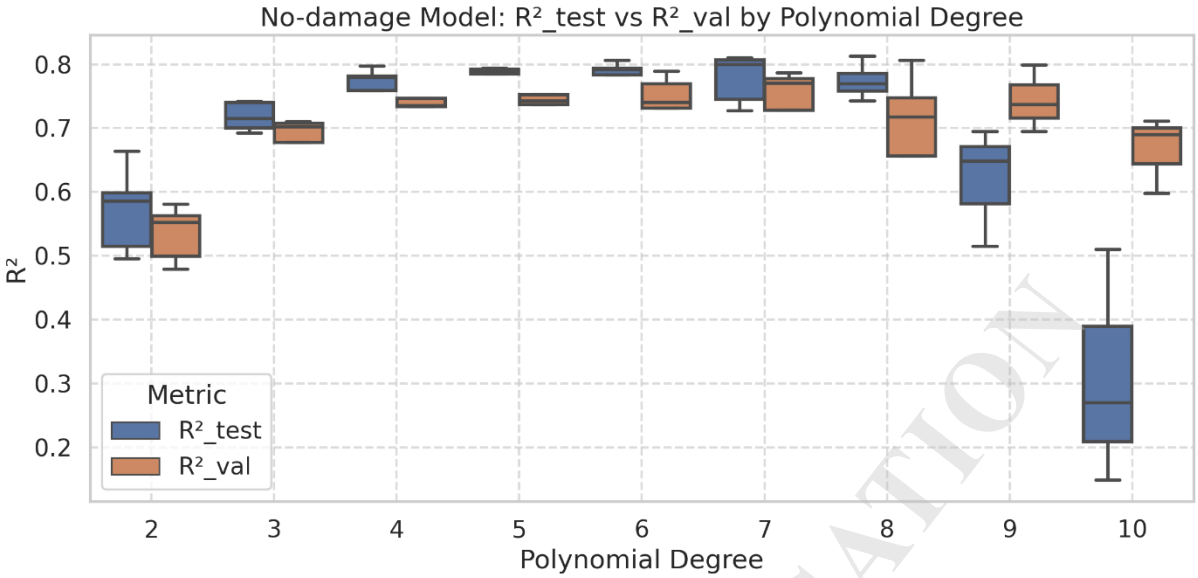


Figure 22. Polynomial Regression degree selection for *LHS5000* database in no Damage Model.

Table 6 shows the results of creating models for different databases. The behavior seems to be quite similar to previous metamodel techniques.

Table 6. Polynomial Regression Results for different databases.

PNR with FF2 based on databases LHS						
Database	R^2training		R^2validation		R^2test	
LHS	Full	No damage	Full	No damage	Full	No damage
LHS5000	0.796782	0.790415	0.787567	0.791222	0.779578	0.789302
LHS2500	0.806827	0.802475	0.762755	0.831658	0.772246	0.831917
LHS1250	0.772573	0.75779	0.775060	0.761361	0.760191	0.775766
LHS625	0.735603	0.730839	0.723093	0.732130	0.725574	0.747068
LHS312	0.749152	0.629371	0.739849	0.581701	0.758322	0.532763
LHS156	0.760563	0.791089	0.776078	0.588699	0.792646	0.496165
PNR with FF2 based on databases QEA						
Database	R^2		Val Loss		Train Loss	
QEA	Full	No damage	Full	No damage	Full	No damage
QEA1687	0.778242	0.862380	0.799098	0.869001	0.742744	0.873709
QEA625	0.751164	0.845151	0.753742	0.809160	0.757736	0.829696
QEA312	0.769839	0.879970	0.777884	0.868079	0.731634	0.861894
QEA156	0.661729	0.906574	0.550962	0.881134	0.613979	0.931966

It is worth to notice, that degree of polynomial starts to be lower for smaller databases. The creation of the PNR model does not take long time, so it was possible to use the cropped

Gridsearch technique to obtain the model with the best score. The selected degrees for every database are shown in Table 7.

Table 7. PNR degree for the best models.

Database LHS	Polynomial Degree	
	Full	No damage
LHS5000	8	5
LHS2500	9	6
LHS1250	6	3
LHS 625	3	3
LHS 312	3	2
LHS 156	3	2

Database QEA	Polynomial Degree	
	Full	No damage
QEA1687	6	4
QEA625	3	3
QEA312	4	3
QEA156	2	3

After obtaining method for selection of PNR degree. QEA optimisation process was submitted for checking results of optimisation basing only on prediction from metamodel. And results was shown in Table 8.

Table 8. PNR prediction vs. FEM

PNR FF2 – based on *LHS5000*			
Model [R^2]	Result	Individual	FEM Result
F- 0.77 ND- 0.789	0.4466	2.5625,2.9375,2.625	0.4858
F- 0.79 ND- 0.8	0.4377	2.6875,2.9375,2.5	0.427044
F- 0.79 ND- 0.8	0.4282	2.6875,2.9375,2.5625	0.43345
F- 0.78 ND- 0..81	0.4456	2.625,2.9375,2.5625	0.4306
PNR FF2 – based on *QEA1687*			
Model [R^2]	Result	Individual	FEM Result
F- 0.77 ND- 0.84	-0.36	1,1,2.9375	0.41940
F- 0.76 ND- 0.88	0.4233	2.6875,2.9375,2.5625	0.43345
F- 0.78 ND- 0.84	0.4245	2.6875,2.9375,2.5625	0.43345
F- 0.73 ND- 0.87	0.4191	2.625,2.9375,2.5625	0.4306

F- 0.76 ND- 0.86	- 11.375	2.9375,2.9375,1	325.659

Although several PNR predictions were close to the FEM results, the method showed clear instability, including failed attempts and non-physical extrapolation errors. For that reason, it is recommended to implement additional mechanism to summitting multiple trials with selection of reliable results, or avoid PNR for different metamodel technique. It would be good approach to develop this approach in future works.

3.4 Results of optimization with selected metamodel

The main purpose of this paper was to present the results of QEA optimisation supported by metamodels. Considering the results obtained within the present evaluation setup, ANN was selected as the most practical metamodel for the subsequent hybrid tests. Although Kriging and PNR also produced competitive results in selected cases, the comparison was not based on a fully uniform repetition protocol; therefore, these observations should be treated as indicative rather than statistically conclusive. The HYBRID1 and HYBRID2 approaches were tested. The minimum threshold for starting the HYBRID approach was set to 300 individuals, based on earlier evaluations of prediction efficiency for smaller datasets. The results for the HYBRID1 approach are shown in Table 9. Four optimisation runs were performed. Only the first run started with an empty database, whereas the subsequent runs were warm-started from progressively enlarged databases containing previously computed FEM evaluations.

Table 9. Results of the HYBRID1 approach.

HYBRID 1 Full Task FF2				
Attempt	Result	Individual	Initial data quantity	Calculation Time
1	0.404056	2.84375 2.96875 2.53125	0	29h
2	0.404975	2.75 2.96875 2.4375	345	3h 30 min
3	0.404056	2.84375 2.96875 2.53125	360	3h 40 min
4	0.404056	2.84375 2.96875 2.53125	377	3h 15 min

This result appears to be quite efficient; however, some interesting phenomena occurred throughout the entire optimisation attempt, as illustrated in Figure 23. In particular, fluctuations in the results are clearly visible.

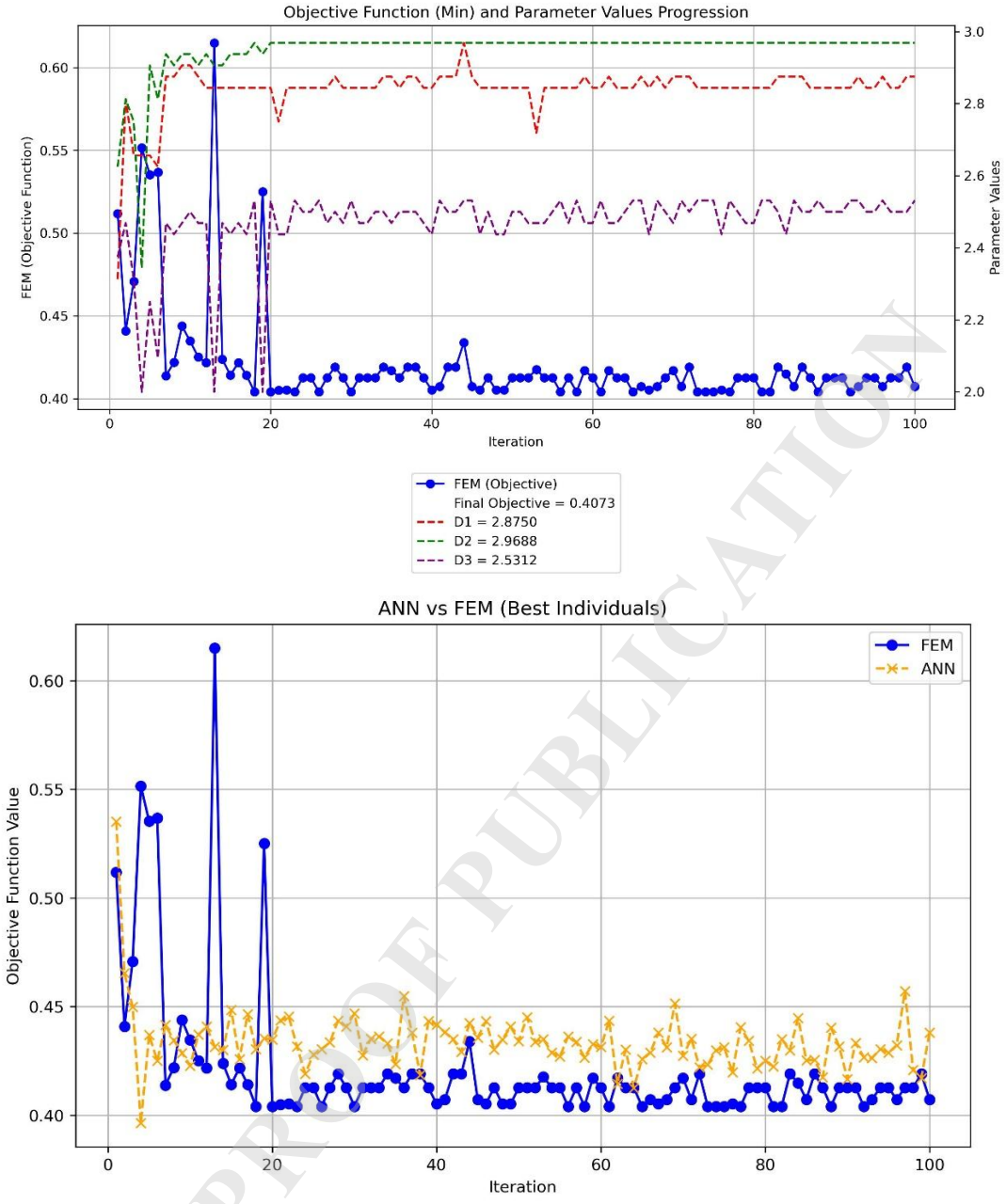


Figure 23. Example optimisation history from one HYBRID1 run.

This behavior is likely caused by the fact that the same minimum fitness function value can correspond to different function outputs for the same input vector. Additionally, the optimisation algorithm explores regions of the design space based on directions suggested by the metamodel. When a result is identified as a global minimum, it indicates that the fitness function value is the lowest observed across all evaluated solutions.

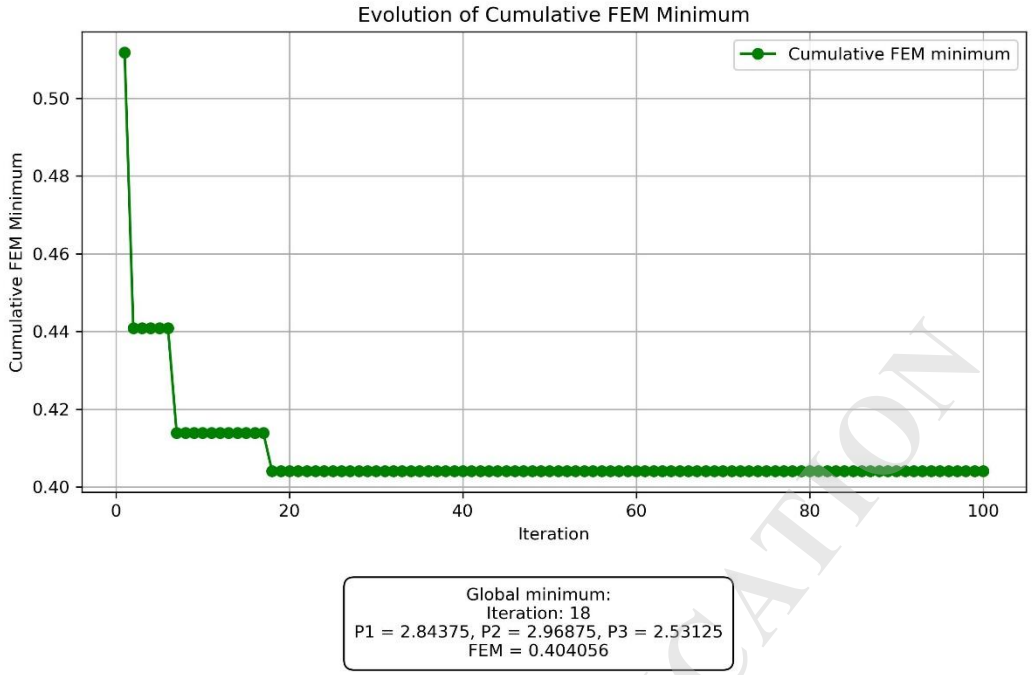


Figure 24. Global minimum of 4th attempt of HYBRID1 approach.

The purpose of evaluating the HYBRID2 approach was to examine whether omitting certain unnecessary calculations could lead to faster results and improve the consistency and stability of the results. The results of these attempts are presented in Table 10.

Table 10. Results of the HYBRID 2 approach.

HYBRID 2 FF2 Full task				
Attempt	Result	Individual	Initial data quantity	Calculation Time
1	0.404975	2.75 2.96875 2.4375	0	25h
2	0.442153	2.625 2.84375 2.34375	307	1h 5 min
3	0.404975	2.75 2.96875 2.4375	313	40 min
4	0.404975	2.75 2.96875 2.4375	316	1h 20 min

In the reported runs, HYBRID2 showed a steadier optimisation course. However, due to the limited number of runs and the warm-started character of runs 2-4, this observation should be treated as qualitative rather than statistically conclusive. The best-performing individual identified in this approach was slightly inferior to that found using the HYBRID1 method.It should also be noted that the best individuals repeatedly approached the upper bound of parameter $P2$. In practical terms, this indicates that, within the present damage-oriented formulation and within the assumed manufacturable thickness range, increasing this wall-thickness parameter remained beneficial for impact resistance. Therefore, the obtained optimum should be interpreted as a boundary-constrained result for the analysed housing concept, not as a general mass-efficiency trade-off.

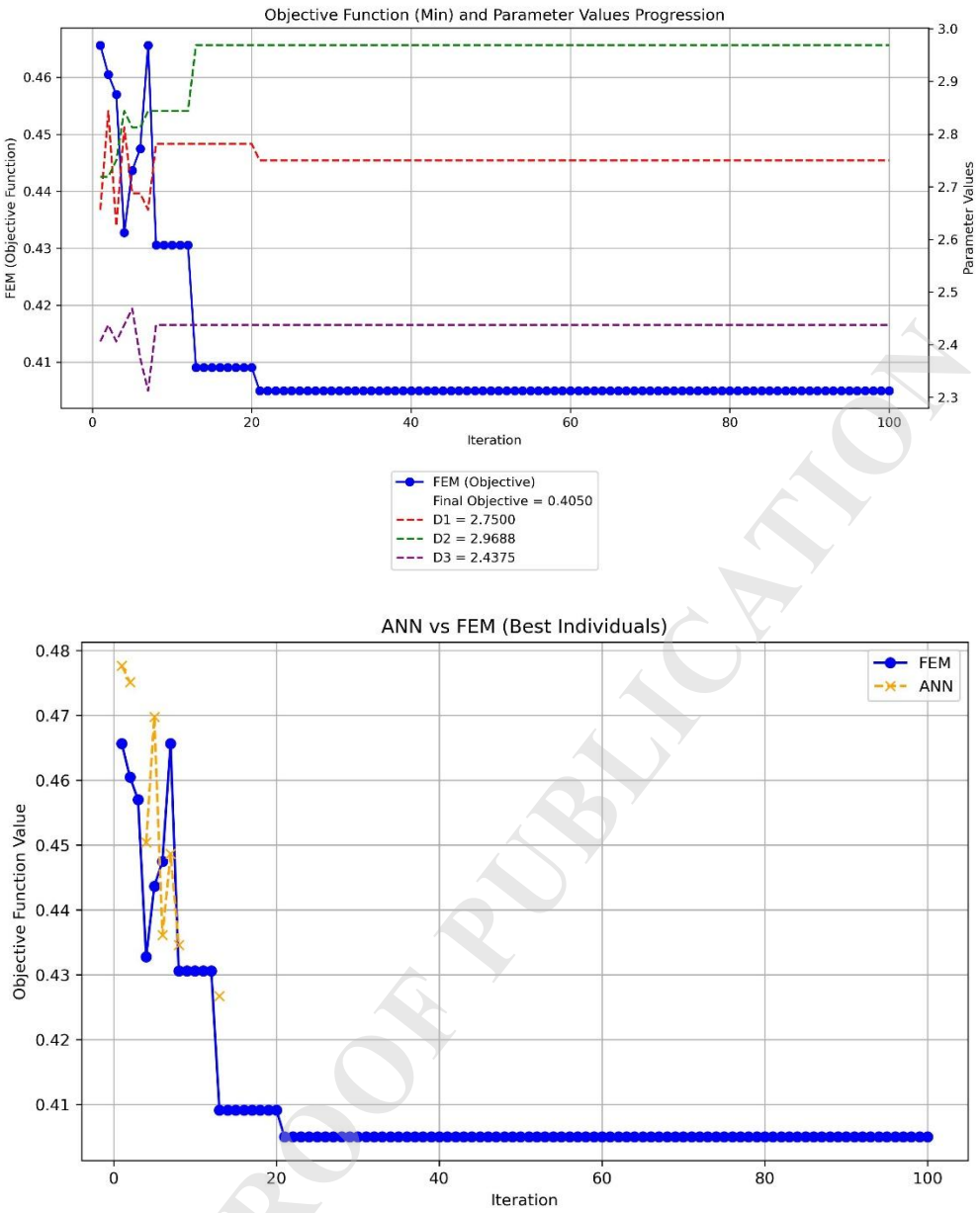


Figure 25. Example of optimization process of HYBRID 2.

4. Conclusions

The presented hybridization of QEA with metamodels significantly reduces computational cost in the task of multiple crash simulations of battery enclosures, while maintaining the quality of design decisions. Compared to the baseline QEA+FEM (approximately 40 hours per single run and ~150 hours for a full series), integrating metamodels shortens the time considerably. Among the tested metamodels, ANN showed the most promising overall behavior within the present study. Kriging provided comparable average prediction quality in several cases, whereas PNR showed instability in selected runs. Because the evaluation protocol was not fully uniform

across all metamodels, these observations should be treated as indicative rather than as a definitive statistical ranking.

The objective function design played a crucial role: variant FF2, through logarithmic transformation of the volume-related term for removed elements, reduced the range of values and alleviated the issue of discreteness near the "damage/no-damage" threshold. This resulted in higher R^2 on smaller datasets and justified its use under iterative data enrichment. The use of two ANN models ("full" and "no-damage"), with switching based on a damage threshold, improved fitting in the relevant design region and stabilized learning—particularly beneficial in the transition zone.

The comparison of hybrid strategies indicates a trade-off between objective value and runtime. In the reported runs, HYBRID1 yielded the lowest objective values, whereas HYBRID2 shortened runtime and showed a steadier optimisation course. Due to the limited number of runs and the warm-started setup, these differences should be interpreted as indicative for the analyzed workflow rather than as statistically conclusive superiority of one stochastic variant over the other.HYBRID1, which enforced FEM verification of the best individual in each phase B iteration and frequent retraining of the network, delivered the best outcomes (objective values ~0.404–0.405 across four runs), although with some performance fluctuations. Its runtime dropped from ~29 hours for an initial empty run to ~3–4 hours for restarts using a pre-collected base of 345–377 individuals. HYBRID2, with fewer mandatory FEM recalculations and less frequent retraining, offered a more stable performance and reduced runtime to ~40–80 minutes when resumed from a base of 307–316 individuals, albeit with a slightly weaker best individual (e.g., 0.404975 vs 0.404056). The minimum threshold to activate hybrid mode was pragmatically set at ≥300 individuals, based on observed prediction reliability on smaller sets. Since the optimizer-level comparison in this study is limited to QEA+FEM and QEA-based hybrid variants, the following conclusions are restricted to the QEA framework.

Within the tested QEA-based framework and for the analyzed battery-housing problem, the combination of QEA, ANN with two networks ('full/no-damage'), and the FF2 objective function was the most useful of the tested variants. These conclusions refer to the present workflow and test case, and should not be interpreted as a general ranking of surrogate-assisted stochastic optimisation methods. Further improvements may involve incorporating uncertainty estimation and active sampling, formalizing the switching criteria between "full" and "no-damage," extending validation with rank-based metrics and absolute errors, and either enhancing the robustness of PNR or replacing it with an ANN/Kriging combination in production scenarios. In summary, for the studied optimization problem, hybridizing the existing QEA workflow with ANN significantly reduces FEM computational cost while maintaining decision quality and repeatability at a practically useful level.

It should be emphasized that the conclusions drawn in this study are based on a single, representative engineering problem. While the obtained results demonstrate the effectiveness of the proposed hybrid optimisation framework, further validation on different types of

structures, loading conditions, and design spaces is required to fully assess its general applicability.

Bibliography

[1] P. Sebastjan, W. Kuś, Method for Parameter Tuning of Hybrid Optimization Algorithms for Problems with High Computational Costs of Objective Function Evaluations, *Applied Sciences*, **13**(10), p. 6307, 2023. https://doi.org/10.3390/app13106307

[2] P. Sebastjan, W. Kuś, For forged automotive components using hybrid optimization techniques, *Computer Methods in Materials Science*, **21**(2), pp. 63–74, 2021. https://doi.org/10.7494/cmms.2021.2.0746

[3] T. Burczyński, W. Kuś, W. Beluch, A. Długosz, A. Poteralski, M. Szczepanik, *Intelligent Computing in Optimal Design*, Springer International Publishing, Cham, 2020. https://doi.org/10.1007/978-3-030-34161-9

[4] J. Jarosz, A. Długosz, Shape Optimization of the Muffler Shield with Regard to Strength Properties, *Engineering Transactions*, **71**(3), pp. 351–366, 2023. https://doi.org/10.24423/ENGTRANS.3093.20230727

[5] H.-S. Park, X.-P. Dang, Structural optimization based on CAD–CAE integration and metamodeling techniques, *Computer-Aided Design*, **42**(10), pp. 889–902, 2010. https://doi.org/10.1016/j.cad.2010.06.003

[6] I. C. Fatiha, S. P. Santosa, D. Widagdo, A. N. Pratomo, Design Optimisation of Metastructure Configuration for Lithium-Ion Battery Protection Using Machine Learning Methodology, *Batteries*, **10**(2), p. 52, 2024. https://doi.org/10.3390/batteries10020052

[7] S. P. S. Rajput, S. Datta, A review on optimization techniques used in civil engineering material and structure design, *Materials Today: Proceedings*, **26**, pp. 1482–1491, 2020. https://doi.org/10.1016/j.matpr.2020.02.305

[8] W. Kuś, W. Mucha, Memetic Inverse Problem Solution in Cyber-physical Systems, [in:] *Advances in Technical Diagnostics*, Springer International Publishing, Cham, 2018, pp. 335–341. https://doi.org/10.1007/978-3-319-62042-8_30

[9] A. J. Keane, J. P. Scanlan, Design search and optimization in aerospace engineering, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **365**(1859), pp. 2501–2529, 2007. https://doi.org/10.1098/rsta.2007.2019

[10] AI and Machine Learning in Simulation, Dassault Systèmes, 2024. Available: https://www.3ds.com/products/simulia/ai-and-machine-learning-simulation

[11] Ansys AI - AI-Augmented Simulation Technology. Available: https://www.ansys.com/ai

[12] AI and Machine Learning for Engineering Design Generation and Optimization, Default. Available: https://altair.com/resource/ai-and-machine-learning-for-engineering-design-generation-and-optimization

[13] C. Ma, C. Xu, M. Souri, E. Hosseinzadeh, M. Jabbari, Multi-Objective Optimisation of the Battery Box in a Racing Car, *Technologies*, **12**(7), p. 93, 2024. https://doi.org/10.3390/technologies12070093

[14] W. Mucha, Application of dynamic condensation for model order reduction in real-time hybrid simulations, *Meccanica*, **58**(7), pp. 1409–1425, 2023. https://doi.org/10.1007/s11012-023-01675-0
[15] R. Jin, W. Chen, T. W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, *Structural and Multidisciplinary Optimization*, **23**(1), pp. 1–13, 2001. https://doi.org/10.1007/s00158-001-0160-4
[16] A. Rurański, W. Kuś, The Quantum-Inspired Evolutionary Algorithm in the Parametric Optimization of Lithium-Ion Battery Housing in the Multiple-Drop Test, *Batteries*, **10**(9), p. 308, 2024. https://doi.org/10.3390/batteries10090308
[17] Y. Bai, T. Wierzbicki, A new model of metal plasticity and fracture with pressure and Lode dependence, *International Journal of Plasticity*, **24**(6), pp. 1071–1096, 2008. https://doi.org/10.1016/j.ijplas.2007.09.004
[18] A. Dean, M. Morris, J. Stufken, D. Bingham, Latin Hypercubes and Space-Filling Designs, Chapman and Hall/CRC, 2015, pp. 613–646. https://doi.org/10.1201/b18619-27
[19] H. Zhu, L. Liu, T. Long, L. Peng, A novel algorithm of maximin Latin hypercube design using successive local enumeration, *Engineering Optimization*, **44**(5), pp. 551–564, 2012. https://doi.org/10.1080/0305215X.2011.591790
[20] H. Zhong, M. L. Brandeau, G. E. Yazdi, J. Wang, S. Nolen, L. Hagan, W. W. Thompson, S. A. Assoumou, B. P. Linas, J. A. Salomon, Metamodeling for Policy Simulations with Multivariate Outcomes, *Medical Decision Making*, **42**(7), pp. 872–884, 2022. https://doi.org/10.1177/0272989X221105079
[21] B. Iooss, L. Boussouf, V. Feuillard, A. Marrel, Numerical studies of the metamodel fitting and validation processes, *International Journal on Advances in Systems and Measurements*, **3**(1 & 2), pp. 11–21, 2010.
[22] K. Kalita, S. Chakraborty, S. Madhu, M. Ramachandran, X.-Z. Gao, Performance Analysis of Radial Basis Function Metamodels for Predictive Modelling of Laminated Composites, *Materials*, **14**(12), p. 3306, 2021. https://doi.org/10.3390/ma14123306
[23] Novel Tool for Selecting Surrogate Modeling Techniques for Surface Approximation, Elsevier, 2021, pp. 451–456. https://doi.org/10.1016/B978-0-323-88506-5.50071-1
[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A Next-generation Hyperparameter Optimization Framework, [in:] *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, Anchorage AK USA, 2019, pp. 2623–2631. https://doi.org/10.1145/3292500.3330701
[25] S. Lee, S. Park, T. Kim, Q. X. Lieu, J. Lee, Damage quantification in truss structures by limited sensor-based surrogate model, *Applied Acoustics*, **172**, p. 107547, 2021. https://doi.org/10.1016/j.apacoust.2020.107547
[26] M. Torzoni, A. Manzoni, S. Mariani, A multi-fidelity surrogate model for structural health monitoring exploiting model order reduction and artificial neural networks, *Mechanical Systems and Signal Processing*, **197**, p. 110376, 2023. https://doi.org/10.1016/j.ymssp.2023.110376
[27] M. A. K. Raiaan, S. Sakib, N. M. Fahad, A. A. Mamun, Md. A. Rahman, S. Shatabda, Md. S. H. Mukta, A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks, *Decision Analytics Journal*, **11**, p. 100470, 2024. https://doi.org/10.1016/j.dajour.2024.100470
[28] S. Hanifi, A. Cammarono, H. Zare-Behtash, Advanced hyperparameter optimization of deep learning models for wind power prediction, *Renewable Energy*, **221**, p. 119700, 2024. https://doi.org/10.1016/j.renene.2023.119700

[29] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, Design and Analysis of Computer Experiments, *Statistical Science*, **4**(4), 1989. https://doi.org/10.1214/ss/1177012413
[30] M. Hutter, The Loss Rank Principle for Model Selection, N. H. Bshouty, C. Gentile, Eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 589–603. https://doi.org/10.1007/978-3-540-72927-3_42
[31] B. Wang, X. Ding, F.-Y. Wang, Determination of polynomial degree in the regression of drug combinations, *IEEE/CAA Journal of Automatica Sinica*, **4**(1), pp. 41–47, 2017. https://doi.org/10.1109/JAS.2017.7510319
[32] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, [in:] Proceedings of the International Conference on Learning Representations (ICLR) 2015, 2015.